REMARKS

Claims 1-8, 14, 16-19, 31, and 32 are pending. Claims 1-8, 14-19, 31, and 32 are rejected. Claim 15 is canceled. Claims 1, 4-7, 14, 16, 18, 19, and 31 are amended. No new matter is added.

Section 112, Second Paragraph Rejection

Claim 31 is rejected because of a lack of previous mention of an array. Applicant has amended claim 31, and respectfully submits that claim 31 particularly points out and distinctly claims what the Applicant regards as the invention. Hence, Applicant respectfully requests that the Section 112 rejection of claim 31 be withdrawn.

Section 102 Rejection

Claims 1-8 and 31 are rejected under 35 U.S.C. §102(b) as being anticipated by Killian et al. (U.S. Patent No. 5,420,992) ("Killian"). Applicant respectfully traverses this rejection.

Applicant respectfully submits that Killian does not disclose or suggest a processor as recited in claim 1. For example, Killian does not disclose or suggest a multi-byte aligned branch instruction having a start address that is an integer multiple, greater than one, of a byte aligned address and this branch instruction is operable to access an instruction at a byte aligned address. Rather, Killian discloses that an m-bit instruction is redefined to operate on N-bit entities and m is less than N. Killian also discloses a program counter that is 16 bits wide in prior art and is 32 bits wide in an extended architecture. There is no disclosure or suggestion in Killian of the multi-byte aligned branch instruction that is operable to access an instruction at a byte aligned address.

To illustrate, Killian does not disclose or suggest "the multi-byte aligned branch instructions are operable to access the plurality of instructions at byte aligned addresses, wherein each of the multi-byte aligned branch instructions has a start address that is an integer multiple of a byte aligned address and the integer multiple is greater than one" as called for by claim 1. An example of this recitation is provided on page 6, lines 14-22; page 7, line 25-page 8, line 3; page 11, line 21-23; and page 11, line 25-page 26, line 11 as follows:

Because instructions are fixed length, the instructions can be aligned with particular memory addresses. For example, instruction memory is addressed in units of bytes. Any instruction having a start address at any byte address is referred to herein as a byte aligned instruction. Any instruction that has a start address that is an integer multiple of a byte address greater than one is referred to herein as a multiple byte aligned instruction. For example, a specific case of a multiple byte aligned instruction is a word aligned instruction. The word aligned instruction has a start address that is aligned at intervals of four bytes . . .

For example, when smaller instructions such as 16-bit instructions are introduced into a 32-bit instruction set, the alignment restrictions are typically reduced to match the size of the smallest instruction. Using the techniques and mechanisms of the present invention, the branch instruction will be able to jump to these instructions even if the new instructions are not multi-byte aligned. For example, if instructions in instruction set were previously word aligned, the new instructions can be byte aligned or half word aligned and the branch instruction will still be able to jump to the appropriate memory location . . .

Techniques and mechanisms of the present invention allow the units of branch instructions to be in bytes even though many of the instructions are multiple bytes in size and a required to be aligned on multi-byte addresses.

Figure 4 is a diagrammatic representation showing a branch instruction that allows byte aligned addresses . . . The branch offset is included in the immediate field 405. However, unlike a conventional branch instruction, the immediate field value is maintained in units of bytes instead of units of words. For example, a conventional instruction would indicate that an offset is thirty-two words in size. The value of 32 would be held in a

conventional branch instruction immediate field. Multiplication by four would indicate that the offset is 128 bytes in size. According to various embodiments, 128 is held in the immediate field 405 instead of the value 32. Because the value of 128 is held, no multiplication or other conversion is needed. The target address is calculated by sign extending the offset to the number of bits in the instruction address and adding it to the branch instruction address or program counter. According to various embodiments, the sign extended offset is added to a variation on the program counter value such as the program counter plus four. In one embodiment, the calculation is as follows:

branch_target = program_counter + sign_extend (immediate_field)

No multiplication by four is required because the units of the branch are bytes and the units of the program counter value and branch target are bytes.

To the contrary, Killian provides in column 2, lines 16-33 that:

[t]he data word size for integer operations is extended from m bits to N bits by widening the machine registers and data paths from m bits to N bits and sign-extending entities of m or fewer bits to N bits when they are loaded into registers. In this context, the term "sign-extending" means writing the most significant bit (the sign bit) of the m-bit entity into the (N-m) most significant bit positions (otherwise undefined) of the N-bit container.

A first subset of the extended architecture instruction set includes the instructions from the previous architecture. These instructions, called m-bit instructions, are redefined to operate on N-bit entities, which may be N-bit sign-extended versions of m-bit (or smaller) entities. For compatibility, the m-bit instructions, when operating on N-bit sign-extended versions of m-bit entities,

must produce an N-bit result that is the N-bit sign-extended version of the correct m-bit result.

Accordingly, Killian discloses that an m-bit instruction is redefined to operate on N-bit entities, which may be sign-extended versions of m-bit entities.

Moreover, Killian describes an address unit (AU) and an execution unit (EU) 15 within a processor. Killian discloses that "the AU includes a program counter (PC) 50 and shares register file 42 with EU 15. In the prior art architecture and hardware implementation, the PC and data paths in the AU are 32 bits wide; in the extended architecture and hardware implementation, they are 64 bits wide" (col. 7, lines 13-21). Thus, Killian discloses a program counter that is 32 bits wide in prior art architecture and is 64 bits wide in prior art architecture.

The disclosure of an m-bit instruction that is redefined to operate on N-bit entities and the disclosure of the program counter that is 32 bits or 64 bits does not disclose or suggest a multi-byte aligned branch instruction having a start address that is an integer multiple, greater than one, of a byte aligned address. The branch instruction is operable to access an instruction at a byte aligned address. Hence, for at least these reasons, claim 1 would not have been anticipated over Killian.

The Office Action suggests in paragraph 8 that the recitation of "substantially all multi-byte branch instructions are operable to access all instructions at byte aligned addresses" is disclosed by column 11, lines 62-64 of Killian. Applicant respectfully disagrees. Applicant respectfully submits that column 11, line 62-64 of Killian does not disclose a byte aligned address as called for by claim 1. Rather, column 11, lines 62-64 discloses a program counter (PC) and a 16 bit offset field. Specifically, column 11, lines 62-64 of Killian discloses that a "branch adder combines the content of the PC and an offset derived from the 16-bit immediate filed in the branch instruction". Further, as disclosed in column 5, lines 23-24, the term 'word' refers to a 32-bit entity. Hence, the disclosure of the program counter and the 16-bit immediate field does not disclose or suggest the byte aligned address as called for by claim 1. Hence, for at least these reasons, claim 1 would not have been anticipated by Killian.

Additionally, the presently pending dependent claims 2-8 and 31 would have been anticipated by Killian because they depend upon independent claim 1.

Hence, for at least the reasons set forth above, Applicants respectfully request that the Section 102 rejection of claims 1-8 and 31 be withdrawn.

Section 103 Rejection

Claims 14-19 and 32 are rejected under 35 U.S.C. §103(a) as being unpatentable over Killian in view of Wittig et al., "OneChip: An FPGA Processor with Reconfigurable Logic" ("Wittig"). Applicant respectfully traverses.

For at least the same reasons set forth above, Killian does not disclose or suggest "each of the plurality of branch instructions and non-branch instructions has a multi-byte length, has a start address that is an integer multiple of a byte aligned address, and is operable to access the plurality of instructions at byte aligned addresses, wherein the integer multiple is greater than one" as called for by claim 14. Moreover, Wittig is not cited to cure the deficiencies stated above in Killian. Rather, as stated in paragraph 19 of the Office Action, Wittig is cited to teach a field programmable gate array. Hence, for at least these reasons, claim 14 would not have been obvious over the combination of Killian and Wittig.

Additionally, the presently pending dependent claims 16-19 and 32 would have been obvious over the combination of Killian and Wittig since they depend upon independent claim 14.

Hence, for at least the reasons set forth above, Applicants respectfully request that the Section 103 rejection of claims 14-19 and 32 be withdrawn.

Conclusion

Accordingly, it is believed that this application is now in condition for allowance and an early indication of its allowance is solicited. Should the Examiner believe that a telephone conference would expedite the prosecution of this application; the undersigned can be reached at the telephone number set out below.

Please charge any required fees or credit any over payments to Weaver Austin Villeneuve Sampson LLP deposit account 504480.

Respectfully submitted,

Nishitkumar V. Patel
Reg. No. 65,546
Weaver, Austin, Villeneuve, and Sampson LLP
P.O. Box 70250
Oakland, CA 94612-0250
(510) 663-1100

Respectfully submitted,

/ G. Audrey Kwan /
G. Audrey Kwan
Reg. No. 46,850
Weaver, Austin, Villeneuve, and Sampson LLP
P.O. Box 70250
Oakland, CA 94612-0250
(510) 663-1100